8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineeering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

SCALABLE TECHNIQUES FOR PARALLEL IMPLEMENTATION OF ADAPTIVE SPACETIME DISCONTINUOUS GALERKIN METHODS

*** Aaron K. Becker[1], Robert B. Haber[2] and Laxmikant V. Kale[1]**

[1] Department of Computer Science
University of Illinois at Urbana-Champaign
201 North Goodwin Ave., Urbana, IL 61801 USA
{abecker3, kale}@uiuc.edu; www.cpsd.uiuc.edu

[2] Mechanical Science & Engineering
University of Illinois at Urbana-Champaign
1206 West Green St., Urbana, IL 61801 USA
r-haber@uiuc.edu; www.cpsd.uiuc.edu

Key Words: *Parallel, Adaptive, Asynchronous, Finite Element, Spacetime, Discontinuous Galerkin, Dynamic Load Balancing, Charm++, ParFUM, Processor Virtualization.*

ABSTRACT

The spacetime discontinuous Galerkin (SDG) method for hyperbolic problems [1] supports an advancing-front solution algorithm that interleaves unstructured spacetime meshing with a patch-wise-local finite element solution procedure. We exploit the local nature of the SDG algorithm to generate spacetime meshes that adapt continuously to capture the trajectories of moving features such as shocks and propagating cracks. Our implementation of the SDG method uses the *Tent Pitcher* meshing algorithm [2] to generate a sequence of patches of spacetime finite elements such that the solution on each patch depends only on adjacent, previously-solved elements and boundary/initial data. We compute the local finite element solution on each new patch as soon as it is generated; therefore, parallelizing Tent Pitcher also parallelizes the SDG finite element solution. We describe here a highly asynchronous parallel implementation of the adaptive Tent Pitcher algorithm, in which each processor can initiate local adaptive operations at unpredictable points in space and time. We base our implementation on the ParFUM framework [3] which in turn is based on the Charm++ runtime system [4]. We associate subdomains of the spatial analysis domain with Charm++ objects whose methods can be invoked remotely. ParFUM manages the distributed data structures needed for parallel execution, with support for asynchronous local adaptive operations and dynamic load balancing.

The SDG algorithm's local and asynchronous nature offers several advantages for achieving a scalable parallel implementation. In conventional time-stepping schemes, the ratio of the average to slowest processor completion times determines the efficiency of processor utilization within each time step, so great care must be taken to ensure a nearly uniform work load per processor per step. Our parallel SDG algorithm does not require global synchronization, and therefore tolerates greater variability in the work load per node. We exploit the SDG algorithm's coarse-grained structure, where patch solution times exceed meshing and communication times by more than an order of magnitude, to further improve performance and scalability. We use object-based virtualization in Charm++ to assign multiple subdomain objects, each with its own thread running Tent Pitcher, to a single physical processor. When one object blocks for communication, another works on patch calculations. The coarse-grained SDG algorithm makes this strategy for hiding communication latency particularly effective.

Our approach to achieving good performance and scalability is closely linked to the capabilities of the Charm++ runtime system. Operations on subdomain boundaries occur irregularly in parallel Tent Pitcher and tend to generate multiple, relatively small messages. This pattern is ill-suited to two-sided communication schemes, but fits well the Charm++ asynchronous message model and benefits from the runtime system's ability to combine small messages automatically for greater efficiency. Local, asynchronous mesh adaptation requires fine-grained locking and careful manipulation of subdomain ghost layers; these are effectively managed by the ParFUM layer and abstracted away from the application code.

Problems with shocks, moving crack-tip fields or other sharp solution features present a significant load balance problem as they generate intense and highly dynamic adaptive mesh refinement. Periodic repartitioning of the spatial domain would be ineffective; these dynamic problems would require frequent repartitioning and the resulting global synchronization would defeat the benefits of our asynchronous algorithm. Instead, we rely on local measures of load imbalance and leverage the virtualization capabilities of Charm++ to overdecompose the problem in order to mitigate the effects of dynamic adaptive mesh refinement. Charm++ offers runtime instrumentation to identify subdomains with too many or too few elements and provides numerous algorithms for migrating subdomain objects between processors to achieve better load balance.

Possible enhancements to our parallel implementation include the possibility of dynamically splitting and merging partitions on the fly to further improve load balance. The large granularity of patch solutions suggests a nested approach to parallelism in which the patch solution is parallelized at the node level, with Tent Pitcher still parallelized across nodes. This approach could leverage available algorithms tuned for multicore performance as well as special-purpose floating-point hardware, such as GPGPUs, without sacrificing the advantages of our current asynchronous parallel algorithm.

Acknowledgment: This work was supported by the Center for Process Simulation & Design, University of Illinois at Urbana Champaign under U.S. National Science Foundation grant DMR-01-21695.

REFERENCES

[1] R. Abedi, S.-H. Chung, M. A. Hawker, J. Palaniappan, and R. B. Haber. "Modeling Evolving Discontinuities with Spacetime Discontinuous Galerkin Methods". In IUTAM Bookseries, Vol. **5**; *Discretization Methods for Evolving Discontinuities*, Springer, 59–87, 2007.

[2] R. Abedi, S. H. Chung, J. Erickson, Y. Fan, M. Garland, D. Guoy, R. B. Haber, J. Sullivan, and Y. Zhou. "Space-time meshing with adaptive refinement and coarsening". In *Proc. 20th Annual ACM Symp. on Comp. Geometry*, 300–309, 2004.

[3] O. Lawlor, S. Chakravorty, T. Wilmarth, N. Choudhury, I. Dooley, G. Zheng, and L. Kale. "ParFUM: A Parallel Framework for Unstructured Meshes for Scalable Dynamic Physics Applications". *Eng. with Comput.*, Vol. **22**, 215–235, 2006.

[4] L. V. Kale, E. Bohm, C. L. Mendes, T. Wilmarth, G. Zheng. "Programming Petascale Applications with Charm++ and AMPI". *Petascale Computing: Algorithms and Applications*, Chapman & Hall CRC Press, 2008.